

UNIVERSITÉ DE MONTRÉAL



CENTRE DE RECHERCHE SUR LES TRANSPORTS



CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

CRT-625

A LINEAR-TIME ALGORITHM FOR
MINIMAX FLOW CENTERS IN TREES

by

Horst A. Eiselt*
Michel Gendreau**

* Department of Business Administration, University of New Brunswick,
P.O. Box 4400, Fredericton, NB E3B 5A3.

** Centre de recherche sur les transports, Université de Montréal, C.P.
6128, Succursale A, Montréal, Québec, H3C 3J7.

Dépôt légal - 2e trimestre 1989

ABSTRACT

In this paper we consider the following problem. Let a tree be given with supplies associated with its vertices. Then the objective is to find a minimax flow center v^* which is defined as a vertex minimizing the largest edge flow which results from shipping all supplies to v^* . We first prove some properties concerning minimax flow centers and then describe a linear-time algorithm for their determination.

Key words : facility location; minimax flow problem; location on trees.

- - - - -

RÉSUMÉ

Étant donné un arbre et des offres d'un produit aux sommets de cet arbre, nous considérons le problème de la détermination d'un centre de flot minimax sur cet arbre, c'est-à-dire d'un sommet tel que si toutes les quantités du produit présentes dans l'arbre sont envoyées à ce sommet, le plus grand flot résultant sur une arête soit minimisé. Nous démontrons d'abord certaines propriétés relatives aux centres de flot minimax, puis nous décrivons un algorithme en temps linéaire pour leur détermination.

Mots-clés : localisation sur les arbres; problèmes de flot minimax; localisation de ressource.

INTRODUCTION

The purpose of this paper is to develop an efficient algorithm for a problem which can be described as follows. Let a tree be given with a positive number, the "supply", associated with each vertex. A facility is to be located at a vertex of the tree, so that the supplies of all vertices are shipped to that facility. The objective is to locate the facility, so that the largest flow on any edge in the associated flow pattern is as small as possible. Such a vertex is called a minimax flow center of the tree.

Applications of this type of model can be found in a variety of situations, e.g. in the military context [3], in communication networks [6] and many others. Recently, Eiselt and Laporte [4] have shown that minimax flow centers are useful in the analysis of some competitive location models in trees.

Some special cases of the problem, viz. minimax flow centers in complete graphs and in trees have been studied by Eiselt and Pederzoli [5]. Their suggested procedure for minimax flow centers in trees requires $O(n^2)$ operations. In this paper, we propose a linear-time algorithm for that problem.

This paper is organized as follows. In the second section, we state and prove a number of properties of minimax flow centers and in the third section, an algorithm is developed and proved. Finally, in the fourth section we outline some possible avenues for future research in this area.

THE PROBLEM AND ITS PROPERTIES

Let a tree $T=(V,E)$ be given with a set of vertices $V = \{v_1, \dots, v_n\}$ and a set of edges $E = \{e_1, \dots, e_m\}$. With each vertex, a positive supply $s(v_i) > 0$, $v_i \in V$ is associated. For any subtree T' of T , the supply of that

subtree is defined as $s(T') = \sum_{v_i \in V'} s(v_i)$. For simplicity, let $s(T)=S$. Given that the facility to which all supplies are shipped is located at v_i , then the associated flow is denoted by $f^i(e_k)$ for all $e_k \in E$ and the flow pattern is $f^i = \{f^i(e_k)\}$. The problem is then to find a vertex v^* and its objective value z^* so as to

$$\left. \begin{array}{l} \text{Minimize } z_i = \max_{e_k \in E} \{f^i(e_k)\} \\ \text{subject to all supplies are shipped to } v_i . \end{array} \right\} \quad (1)$$

Any vertex v^* which solves problem (1) is called a minimax flow center or MMFC. In the following we will prove some properties of problem (1) which lead to an optimal algorithm for the determination of an MMFC.

Suppose that the deletion of edge e_k from E divides T into two subtrees $\bar{T}_1^k$ and $\bar{T}_2^k$. Then we can prove

Lemma 1: $z^* = \max_{e_k \in E} \{\min \{s(\bar{T}_1^k); s(\bar{T}_2^k)\}\} < S/2.$

Proof : Let v^* be an MMFC and let $f^* = \{f^*(e_k)\}$ denote the unique flow pattern induced by locating the facility at v^* . By definition, we have

$$z^* = \max_{e_k \in E} \{f^*(e_k)\} . \quad (2)$$

First we will prove that

$$f^*(e_k) = \min \{s(\bar{T}_1^k); s(\bar{T}_2^k)\} \text{ for all } e_k \in E .$$

For that purpose, consider an arbitrary edge $e_k = (v_1^k, v_2^k)$, so that $v_1^k \in \bar{T}_i^k$, $i=1,2$. Without loss of generality, let $v^* \in \bar{T}_1^k$. As all supplies in $\bar{T}_2^k$ are shipped to v^* via e_k , we obtain

$$f^*(e_k) = s(\bar{T}_2^k) . \quad (3)$$

If it can be shown that

$$s(\bar{T}_2^k) < s(\bar{T}_1^k) , \quad (4)$$

then (2), (3) and (4) imply the first part of the relation in lemma 1. In order to do so, suppose that $s(\bar{T}_2^k) > s(\bar{T}_1^k)$ and consider the flow pattern f' that would result from locating the facility at vertex v_2^k . Then the following relations hold:

$$f'(e_\ell) < s(\bar{T}_1^k) \text{ for all } e_\ell \text{ in } \bar{T}_1^k ,$$

$$f'(e_k) = s(\bar{T}_1^k) ,$$

which, as $s(\bar{T}_1^k) < s(\bar{T}_2^k)$ is assumed, implies

$$f'(e_\ell) < s(\bar{T}_2^k) \text{ for all } e_\ell \text{ in } \bar{T}_1^k .$$

Furthermore, $f'(e_\ell) < s(\bar{T}_2^k) - s(v_2^k)$ for all e_ℓ in $\bar{T}_2^k$, which, with $s(v_2^k) > 0$, implies $f'(e_\ell) < s(\bar{T}_2^k)$ for all e_ℓ in $\bar{T}_2^k$.

Therefore, $f'(e_\ell) < s(\bar{T}_2^k) < z^*$ for all $e_\ell \in E$ which contradicts that v^* is an MMFC.

Thus

$$f^*(e_k) = \min \{s(\bar{T}_1^k); s(\bar{T}_2^k)\} \quad \text{for all } e_k \in E. \quad (5)$$

Note that for any $e_k \in E$, $s(\bar{T}_1^k) + s(\bar{T}_2^k) = S$ which implies

$$\min \{s(\bar{T}_1^k); s(\bar{T}_2^k)\} < S/2 \quad \text{for all } e_k \in E. \quad (6)$$

From (2), (5) and (6) we may conclude that

$$z^* = \max_{e_k \in E} \{ \min \{s(\bar{T}_1^k); s(\bar{T}_2^k)\} \} < S/2. \quad \blacksquare$$

Suppose now that vertex v_i has a degree of $|E_i|$, so that its removal from V along with all edges incident to it generates subtrees $T_1^i, T_2^i, \dots, T_{|E_i|}^i$. Then we can state

Lemma 2: The vertex v_i is an MMFC if and only if

$$s(T_j^i) < S/2, \quad j = 1, \dots, |E_i| \quad (7)$$

Proof : Let e_j^i , $j = 1, \dots, |E_i|$ denote the unique edge connecting T_j^i to v_i . Locating the facility at v_i implies that

$$f^i(e_j^i) = s(T_j^i), \quad j = 1, \dots, |E_i|,$$

and

$$f^i(e_\ell) < s(T_j^i), \quad \text{for all } e_\ell \in T_j^i, \quad j = 1, \dots, |E_i|,$$

and thus

$$\max_{e_k \in E} \{f^i(e_k)\} = \max_j \{s(T_j^i)\}. \quad (8)$$

First we prove the necessity of condition (7). If (7) is not satisfied, then there exists exactly one index j such that $s(T_j^1) > S/2$ and thus $f^1(e_j^1) > S/2$. According to lemma 1, v_j cannot be an MMFC.

We now prove the sufficiency of (7) by contradiction. Let condition (7) be satisfied but assume that v_i is not an MMFC. Then there must be a vertex $v' \neq v_i$ which is an MMFC. Without loss of generality, suppose that v' is in T_1^1 . Then

$$\begin{aligned} f'(e_1^1) &= S - s(T_1^1) \\ &> S/2 \end{aligned} \quad \text{[by (7)]}$$

or simply $f'(e_1^1) > S/2$. Then

$$\begin{aligned} z' &= \max_{e_k \in E} \{f'(e_k)\} > f'(e_1^1) \\ &> S/2 \quad \text{[by (9)]} \\ &> s(T_j^1), \quad j = 1, \dots, |E_i| \quad \text{[by (7)]} \\ &> \max_{e_k \in E} \{f^1(e_k)\} \quad \text{[by (8)]} \\ &= z_i \quad \text{[by (1)]} \end{aligned}$$

or $z' > z_i$ and thus v_i must be an MMFC. ■

Corollary 3: The vertex v_i is the unique MMFC if and only if

$$s(T_j^1) < S/2 \quad \text{for all } j = 1, \dots, |E_i|. \quad (10)$$

Proof : Assume that the facility is located at some vertex v' and let (10) be satisfied. If v' is in T_j^1 , then $z' > f'(e_j^1) = S - s(T_j^1) > S/2$ and by virtue of lemma 1, v' cannot be an MMFC.

Suppose now that v_1 is an MMFC but relation (10) is not satisfied, i.e. there exists exactly one index j such that $s(T_j^1) = S/2$. Without loss of generality, let $j=1$. Let the edge e_1^1 connect v_1 and v_1^1 . It is easily verified that at v_1^1 , condition (7) is also satisfied. Thus v_1^1 is also an MMFC. ■

The case of non uniqueness of the MMFC can however be explored further.

Lemma 4: Given that $s(v_1) > 0$, for all $v_1 \in V$, a tree T admits exactly two MMFCs if and only if there exists an edge $e_k = (v_1^k, v_2^k)$, such that $s(T_1^k) = s(T_2^k) = S/2$. The two MMFCs are then v_1^k and v_2^k .

Proof : It is easily seen that v_1^k and v_2^k satisfy the sufficient condition for MMFCs of lemma 2. Consider now a vertex $v' \neq v_1^k, v_2^k$. Without loss of generality let v' be in $\bar{T}_1^k$. Deleting v' and all of its incident edges, a number of subtrees are generated. One of these subtrees contains v_1^k as well as $\bar{T}_2^k$ in its entirety; denote this subtree by T' . Then $s(T') > s(v_1^k) + s(\bar{T}_2^k) > S/2$ as $s(v_1^k) > 0$ and $s(\bar{T}_2^k) = S/2$. Therefore v' cannot be an MMFC and v_1^k and v_2^k are the only two MMFCs in T . ■

AN ALGORITHM FOR FINDING MMFCs IN A TREE

In this section we first describe the basic idea of the algorithm. Then the method is formally described and a few remarks in regard to its implementation are offered. A modification allowing zero supplies at vertices is then delineated. Finally, it is proved that the algorithm solves the problem of finding minimax flow centers and that it does so in linear time.

The proposed algorithm consists of six steps which can be organized in four phases. The first phase (step 0) initializes the method. Phase 1 (steps 1 and 2) organizes the given data set in a way convenient for processing. Phase 2 consists of the iterative steps 3 and 4 and finally phase 3 includes steps 5 and 6 which describe the solution in cases of uniqueness and non uniqueness, respectively. More specifically, in phases 0 and 1 the following data sets are generated. $E(v_i)$, $v_i \in V$, is an edge list consisting of the incidence relations of edges and vertices. The parameters $d(v_i)$, $v_i \in V$, is a listing of the degrees of the vertices of the tree and $\tilde{s}(v_i)$, $v_i \in V$, specifies the modified supplies of all vertices. Initially, the modified supplies of all vertices equal the given supplies $s(v_i)$, $v_i \in V$. The set P consists of all leaves of the current tree, i.e. all vertices with degree one. Finally, the total supply S is computed and z denotes the largest edge flow encountered so far.

The basic idea of the algorithm (phase 2) is this. Consider any edge incident to a leaf in the tree, i.e. some edge $e' = (v', v'')$ with $d(v') = 1$. Then according to lemma 2, v' is an MMFC if $\tilde{s}(v') > S/2$ which is easily checked. If this is the case, v' is an MMFC and we are through. Otherwise, it is known that all supplies available at v' have to be shipped to v'' via e' . Thus the tree can be contracted by deleting v' and e' , noting that $f(e') = s(v')$. Update z if necessary and increase the (modified) supply of v'' by $s(v')$ and decrement the degree of v'' by one. Then v' is deleted from the set P and the procedure is repeated. At some point, the stopping criteria in step 3 lead to the solution which is given in steps 5 and 6.

Formally, the algorithm can be described as follows.

Algorithm: Minimax Flow Center(s) in Trees

Step 0 (Initialization):

Set

$$E(v_i) \leftarrow \emptyset, \text{ for all } v_i \in V,$$

$$P \leftarrow \emptyset,$$

$$d(v_i) \leftarrow 0, \text{ for all } v_i \in V,$$

$$S \leftarrow z \leftarrow 0.$$

Step 1 (Set up vertex data structures from edge data):

For each edge $e_k = (v_1^k, v_2^k)$, $k = 1, \dots, n-1$, set

$$E(v_i^k) \leftarrow E(v_i^k) \cup \{e_k\}, \quad i = 1, 2.$$

$$d(v_i^k) \leftarrow d(v_i^k) + 1, \quad i = 1, 2.$$

Step 2 (Computation of total supply and construction of pendant vertex list P):

For each vertex v_i , $i = 1, \dots, n$, set

$$S \leftarrow S + s(v_i),$$

$$\tilde{s}(v_i) \leftarrow s(v_i),$$

$$P \leftarrow P \cup \{v_i\}, \text{ if } d(v_i) = 1.$$

Step 3 (Pendant vertex processing):

Pick some $v' \in P$.

If $d(v') = 0$, then v' is the last vertex in the tree, go to step 5.

If $d(v') \neq 0$, let $e' = (v', v'')$,

set $P \leftarrow P \setminus \{v'\}$ and

if $\tilde{s}(v') < S/2$, go to step 4,

if $\tilde{s}(v') > S/2$, go to step 5,

if $\tilde{s}(v') = S/2$, go to step 6.

Step 4 (Deletion of a non-optimal vertex):

Set $\tilde{s}(v'') + \tilde{s}(v'') + \tilde{s}(v')$,
 $z + \max \{z, s(v')\}$,
 $E(v'') + E(v'') \setminus \{e'\}$,
 $d(v'') + d(v'') - 1$.

If now $d(v'') = 1$, set $P + P \cup \{v''\}$.

Go to step 3.

Step 5 (A unique MMFC has been found):

Set $z^* + \max \{z, S - \tilde{s}(v')\}$.

Now v' is the unique MMFC and the largest edge flow is z^* .

STOP.

Step 6 (Multiple MMFCs have been found):

Set $z^* + S/2$.

Now v' and v'' are the two MMFCs and the largest edge flow is z^* .

STOP .

As far as the implementation of the above algorithm is concerned, two notes are in order. First, to allow for insertion and deletion of edges into and from edge lists in constant time each, doubly-linked lists should be used. Secondly, it would be convenient to organize list P as a stack.

The algorithm could easily be modified to account for vertices with zero supply. In such a case more than two MMFCs may exist and thus a list V^* of optimal vertices is needed. This could be accomplished by modifying steps 5 and 6 as shown below.

Step 5' (Identification of the last MMFC):

Set $z^* + \max \{z, S - \tilde{s}(v')\}$ and
 $V^* + V^* \cup \{v'\}$.

STOP.

Step 6' (Identification of multiple MMFCs):

Set $V^* = V^* \cup \{v'\}$.

Go to step 4.

We now prove the validity of the algorithm.

Theorem 5: The proposed algorithm determines the MMFCs in a tree and the minimax flow value in $O(n)$ operations.

Proof : We first prove the validity of the algorithm and then count the operations required for its execution. The principle of contraction has already been explained above. Suppose now that $\tilde{s}(v') < S/2$ in step 3 at some time during the iterations. Then in the original tree the edge e' separates two subtree $\bar{T}'_1$ and $\bar{T}'_2$, where currently $\bar{T}'_1$ is represented by v' , so that $\tilde{s}(v') = s(\bar{T}'_1)$. Thus, by assumption $s(\bar{T}'_1) < S/2$ and as $s(\bar{T}'_1) + s(\bar{T}'_2) = S$, $s(\bar{T}'_2) > S/2$ which, according to lemma 2, implies that v' cannot be an MMFC. Thus the algorithm can contract the vertex v' which is done in step 4. If now $\tilde{s}(v') > S/2$, v' must be the unique MMFC since no vertex in $\bar{T}'_2$ can be an MMFC and all other vertices in $\bar{T}'_1$ have already been contracted which implies that they cannot be MMFCs. The case where $\tilde{s}(v') = S/2$ uses lemma 4 concluding that v' and v'' are both MMFCs. Note that the value of z is always updated so that it equals the largest edge flow encountered so far. We may therefore conclude that the algorithm solves the problem.

As far as the operations count is concerned, we can state the following:

- Steps 0, 1 and 2 are executed once each requiring $O(n)$ operations.
- The iterative phase consisting of steps 3 and 4 is executed at most once for each edge, i.e. no more than $(n-1)$ times. Each such execution requires a constant number of operations.

- Steps 5 and 6 require a constant number of operations.

Thus, the algorithm has a worst-case complexity of $O(n)$. As each of the $O(n)$ parameters of the problem has to be considered at least once by any algorithm, $O(n)$ is also a lower bound on the complexity. Hence the algorithm proposed here is optimal except maybe for a constant factor. ■

CONCLUSION

In this paper an optimal algorithm for the determination of minimax flow centers in trees has been devised. Future research could consider two possible extensions. On the one hand, a weight may be associated with each edge, so that the objective is then to find a vertex so as to minimize the largest weighted edge flow in the tree. On the other hand, we may try to find MMFCs in graphs other than trees. A straightforward generalization of the algorithm outlined here does not appear to exist as the flow patterns towards the facility location are no longer unique but require the use of a minimax flow algorithm such as those presented by Ahuja [1,2].

ACKNOWLEDGMENTS

This research was in part supported by the Natural Sciences and Engineering Research Council of Canada under grant numbers A9160 and OGP0038816. This support is gratefully acknowledged.

REFERENCES

- [1] Ahuja, R.K. (1985) "Minimax Linear Programming Problem", Operations Research Letters 4, 131-134.
- [2] Ahuja, R.K. (1986) "Algorithm for the Minimax Transportation Problem", Naval Research Logistics Quarterly 33, 725-739.
- [3] Eiselt, H.A. (1981) "Min-Max and Max-Min Problems in Linear Programming", Proceedings of the ASAC 2/2, Halifax, 13-25.
- [4] Eiselt, H.A., Laporte, G. (1989) "Equilibrium Conditions for the 3-Facility Hotelling Model in a Tree", working paper 89-002, Faculty of Administration, University of New Brunswick, Fredericton, Canada.
- [5] Eiselt, H.A., Pederzoli, G. (1984) "A Location Problem on Graphs", New Zealand Operational Research 12, 49-53.
- [6] Minoux, M. (1976) "Flots équilibrés et flots avec sécurité", E.D.F. Bulletin de la Direction des Études et Recherches - Série C, 5-16.